

# 3D plane-based egomotion for SLAM on semi-structured environment

Diego Viejo and Miguel Cazorla  
Dept. de Ciencia de la Computación e I. A.  
Universidad de Alicante  
P.O. Box 99 03080 Alicante  
{dviejo,miguel}@dccia.ua.es

**Abstract**—Several works deal with 3D data in SLAM problem. Data come from a 3D laser sweeping unit or a stereo camera, both providing a huge amount of data. In this paper, we detail an efficient method to extract planar patches from 3D raw data. Then, we use these patches in an ICP-like method in order to address the SLAM problem. Using ICP with planes is not a trivial task. It needs some adaptation from the original ICP. Some promising results are shown for outdoor environment.

**Index Terms**—3D robot mapping, plane extraction, registration, SLAM.

## I. INTRODUCTION

SLAM is a well-known problem in robotics community. It consists of recovering a spatial map of the environment where an autonomous vehicle or robot is moving on, while it attempts to estimate its own pose (location and orientation) relative to the map. There are a lot of previous works that try to address the problem of SLAM (e.g. [1], [2]).

Our main goal in this paper is to work in semi-structured environment, i.e., man-made indoor and outdoor environment. We use a sweeping unit with a 2D laser Sick as a main sensor, mounted on a PowerBot. It provides 3D data with low error and higher range (compared to stereo systems [3]). However, our main aim is to deal with outliers, i.e., environments with people or not modeled objects. This task is hard to manage because classic algorithms, like ICP [11], are very sensitive to outliers. Furthermore, we will not use odometry information.

As seen in previous works, it is necessary to reduce the huge amount of data provided for the sensor. Instead of handling raw 3D points data, we use a geometric feature extraction approach, as shown below, that gives us planar patches belonging to the surfaces of both indoor and outdoor 3D scene. Then these patches are used in an ICP-like method, which is able to work with planes. Our algorithm is restricted to man-made environments (both indoor and outdoor), because it needs different planar surfaces with at least two main orientations.

In [4] a ICP-based method is used with 3D raw points, to which a saliency detector process is applied in order to reduce the huge amount of data. The same authors propose in [5] a modified version which is able to *close the loop*. Other approaches for 3D mapping use EM algorithm, Hough transform, or grid based algorithms ([6], [7], [8], [9], [10]). Some of them work directly with raw 3D data and others with an intermediate plane or feature extraction process.

The rest of the paper is organized as follows. In Section II we present the data acquisition process and describe the system used. Then, Section III details how we can reduce the huge amount of data, extracting planar patches. Further on, in Section IV we explain in detail how we have modified the original ICP algorithm in order to manage planar patches. Several experiments are described in Section V. We conclude in Section VI with our future work.

## II. DATA ACQUISITION

Data come from a standard 2D Sick laser with a laser sweeping unit. This unit allows to capture 3D data. The robot is driven around its environment. As it moves it gets data from the sweeping unit. The robot takes  $N$  3D observations performed from different poses. Assuming that the robot is confined to the  $XZ$  horizontal plane and also that the data capture system is fixed, each pose can be described as coordinates in the horizontal plane and a rotation around the vertical axis  $Y$ . Given a pose, the next one is obtained through the incremental action  $\mathbf{a}_t = [\delta x_t, \delta z_t, \delta \theta_t]$  between two consecutive observations (3D acquisitions). Figure 2 shows an example of data obtained in both indoor and outdoor environment. Each data set is a 3D point cloud.



Fig. 1. Our mobile robot equipped with laser Sick and a sweeping unit.

The entire 3D scanner is mounted on a PowerBot from Mobile Robotics (see Figure 1). This kind of robot allow us to use it on outdoor scenarios.

## III. MODEL COMPLEXITY REDUCTION

We want to reduce the complexity of a 3D scene formed by a huge set of 3D points. In spite of handling raw 3D points data, we'll perform a study of planar surfaces contained into the 3D scene. Planar surfaces can be used to register the movement done by a robot between two consecutive poses for both indoor and semi-structured outdoor environments

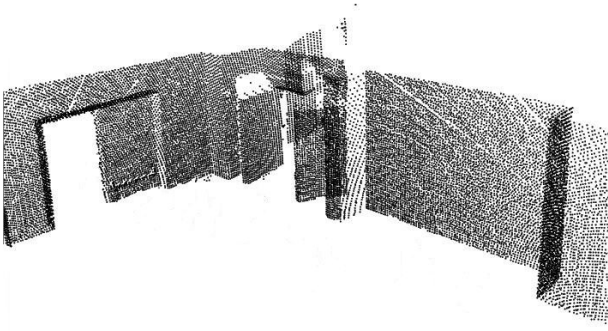


Fig. 2. 3D data from the sweeping unit. Left: indoor environment. Floor and roof points have been removed to improve visualisation. Right: outdoor.

like urban landscapes. Usually, building facades are formed by planar surfaces and we can exploit this feature to perform pose registration. The set of 3D points from an individual scene retrieved by our 3D sweeping unit has about 65,000 points for outdoor scenes up to 100,000 for indoor environments. In this section we propose an approach to reduce scene complexity to a few number of planar patches (less than 200 per scene).

We use the method proposed by Martin, Gomez and Zalama propose in [13] to estimate local surface directions. Principal Component Analysis is performed over a point  $p_i$  and its neighbourhood by mean of a Singular Value Decomposition. This process retrieves the underlying surface normal vector of a given set of points if these fit a planar surface. Furthermore, a threshold called *thickness angle* can be defined from singular values in order to determine in which situations a point, as well as its neighbourhood, belong to a planar surface or not. This *thickness angle* can be used to measure the fitting of a 3D point set to a plane. The lower *thickness angle* we found, the better fitting between points and planar surface is. If this value is low enough, we can assume that all points into a window belong to a plane, and the estimated normal vector will be the normal vector at each point. Thus, we can easily compute the parameters of the planar patch into the window without computing the normal vector for all the points into the 3D scene as we see below. The size of the planar patches depends on the size of the window used to compute the Singular Value Decomposition.

As it is considered in [4], sample density of 3D laser range finder data presents large variations due to the divergence of consecutively sampled beams. If we take constant size neighborhood we can not obtain planar patches at a certain distance where points are too far from each other. So we propose a dynamic size window depending on the distance between a point in the 3D scene and the coordinate origin (viewpoint). Also, we have to ensure a minimum number of points inside the window in order to do reliable the Singular Value Decomposition result. [13] uses a minimum number of points of nine. We know that the angle between two

consecutive sample beams is half degree, so we can compute easily the size of the window, depending on the distance of the point, that ensures at least nine points inside the window.

With this approach we can estimate normal vector direction of the planar patches, but we can not know the orientation of these vectors. Nevertheless, we are trying to find planar patches just in single 3D scene so we have viewpoint information. The correct orientation of a normal vector  $\hat{n}_i$  can be set trivially. Given a viewpoint  $O_j$  we have to flip  $\hat{n}_i$  if  $(\hat{n}_i \cdot \frac{(v_i - p_i)}{\|v_i - p_i\|}) < 0$ . We have no troubles with sharp edges because *thickness angle* avoids normal vector estimation near sharp edges.

Finally, we use an automatic seeded selection algorithm [14] [15] to perform an efficient planar patch extraction from the whole 3D scene. The main criteria to select a point and its estimated normal to be a planar patch will be its *thickness angle* since a point with a low thickness angle is appropriately representative of the points into its neighbourhood. We select and remove points from the 3D scene randomly. A selected point (with its normal vector estimated) is stored in the final seeds vector if this point is not inside the neighbourhood of any other selected seed and if its *thickness angle* is low enough to ensure reliability. At the end of this process, we obtain the planar patches of a 3D scene. We can see some results both from outdoor and indoor scenes in figure 3. Patch boundaries are computed from the size of the window used to estimate normal vector. The resulting 3D models have from 150 patches for indoor to 50 patches for outdoor scenes, therefore we achieve a substantial model complexity reduction.

#### IV. POSE REGISTRATION

Pose registration is a well known topic into computer vision and mobile robots researches. Classical Iterative Closest Point Algorithm [11] (ICP) is widely used for geometric alignment of a pair of three-dimensional points sets. In mobile robotics, these point sets usually come from range sensors of a robot in two consecutive poses. It's assumed that the rigid transformation that aligns the two point sets fits with

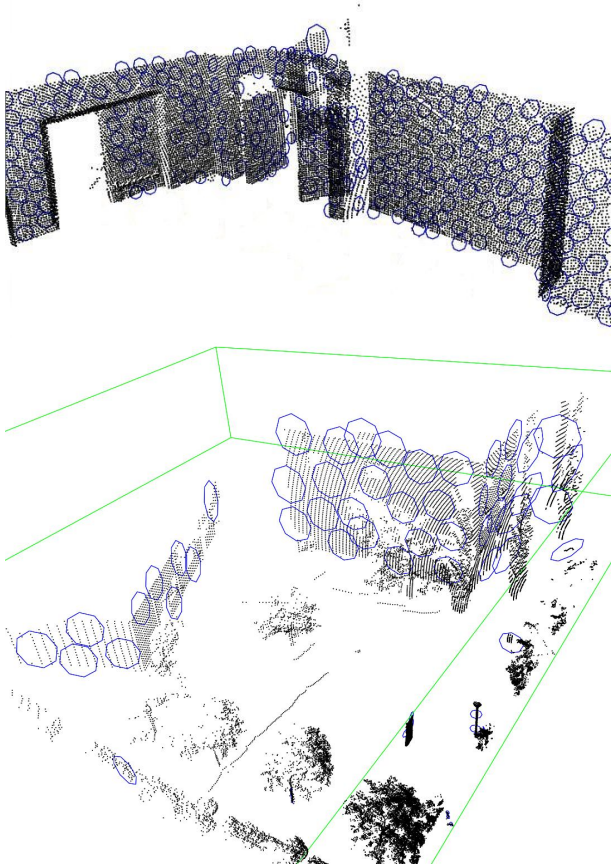


Fig. 3. Planar patches extracted from 3D laser range finder data. Patches are represented by blue circles. Radius of each circle depends on the size of the window used to compute the patch. Top: indoor environment. Bottom: outdoor environment

the movement performed by the robot. Nevertheless, ICP not always reaches a global minimum in presence of outliers. In fact, the movement of the robot itself produces outliers, it is said, some regions in the first pose can not be seen in the second and vice versa. Furthermore, the high amount of data retrieved by 3D laser makes computations harder. In the literature there are a lot of variants of the ICP algorithm that improve its efficiency. Nevertheless, both original algorithm and its variants try to match two point sets. In fact, in [11] Besl and McKay specify that in order to register any two 3D shapes, point set of each shape is needed. In this paper we propose a new approach based on ICP algorithm that register directly two sets of planar patches instead of using two sets of points. The advantage of using planar patches is that we know not only the geometric position of each path but also its orientation given by its normal vector. Thus, performing registration with planes allow us to reduce the complexity of the 3D scenes as we have seen in the previous section.

Summarising, the original ICP algorithm performs the matching between two points set in two steps. The algorithm begins with an initial transformation. This transformation is applied to the first point set and each point of this set

is matched with the closest point in the other set. Next, registration is computed calculating the transformation that fits better the matches calculated in the previous step. This transformation is again applied to the initial point set. These two steps (find closest points and transformation calculation) are calculated iteratively until convergence. Some approaches use a modified version of ICP, calculating all the distances from one point to the entire second set and using the distance to weight each match. This produces a faster and better registration.

In our case, we are going to exploit both the information given by the normal vector of the patch planes and its geometric position. Although both (position and orientation) are related, we have noted experimentally that they can be calculated separately. This is due to the special structure of our environments: man-made buildings with planar structures. We demonstrate in some experiments, both indoor and outdoor, that our approach can be used for performing Egomotion. In this initial work we only consider a 3DOF movement. First, we register the orientation of planar patch sets and then we address the translation registration.

#### A. Orientation Registration

The first step to register the orientation between two plane sets consist of applying a modified version of the original ICP algorithm over the two normal vector sets obtained from the planes estimated from two consecutive 3D scenes. The matching criteria for the first step of ICP could be the angle between normal vectors. The rotation that each match provide to the resulting one is the angle between the normal vectors of the patch planes weighted by the next expression: Let  $\alpha$  be the angle between the vectors of the  $i$ -th match, the weight of  $i$ -th match is computed as

$$w_i = \gamma e^{-\frac{\alpha^2}{\sigma^2}} \quad (1)$$

where variance  $\sigma$  depends on the maximum rotation that the robot can perform *a priori*. High values for  $\sigma$  allow the registration of high rotations in the robot pose, but make harder the achievement of a global minimum. On the other hand, small  $\sigma$  values limit the movement model, but improve the results. In figure 4 we can see the results of applying vector angle distance based ICP. As we can see, this method is prone to match moreover all patches from one set with just one of the patches in the second set. Thus, outliers affects too much in the resulting rotation. We can consider outliers as those matches between patches that has no spatial relationship. We have a 30% of non-successful orientation registration situations.

An improvement of this method consist of using not only the angle between vectors as matching criteria and weighting but also the euclidean distance between patches matched. In this way we want to avoid outliers by reducing its match weight and, as a result, improving matching consistency. Let  $\alpha$  be the angle between the vectors and  $d$  the euclidean distance between the patches of the  $i$ -th match, the weight of this is now computed as:

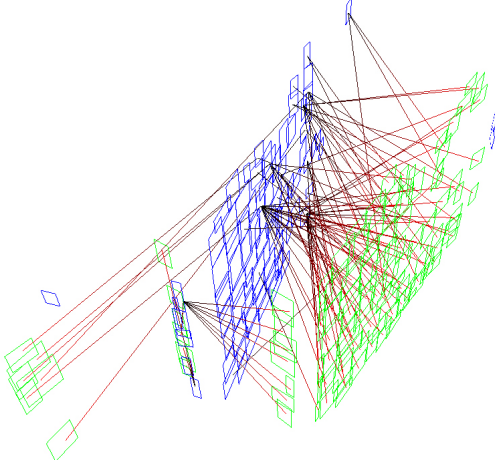


Fig. 4. Planar patch matching just using angle between vectors as distance matching criteria for ICP orientation registration. Patches from each set are colored blue and green respectively. Red lines show matches.

$$w_i = \gamma \left( e^{-\frac{\alpha_1^2}{\sigma_1^2}} + \lambda e^{-\frac{d^2}{\sigma_2^2}} \right) \quad (2)$$

where  $\lambda$  is a constant that allows us to adjust how distances affect the results in order to avoid matches just by distance. Empirically, we have found that  $\lambda = 0.15$  retrieves the best results. In figure 5 we can see how matches have higher consistency than in figure 4. Thus in this situation, outliers almost don't affect results. Now, with this improvement, we have found that the mean of the error empirically measured is always under 0.1 rad.

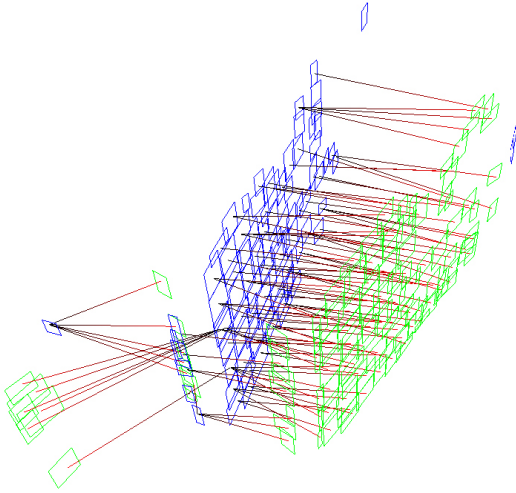


Fig. 5. Planar patch matching using both angle between vectors and Euclidean distance as matching criteria for ICP orientation registration. Patches from each set are colored blue and green respectively. Red lines show matches.

### B. Position Registration

After orientation registration we can assume that the two planar patch sets are aligned respect to the orientation of the patches. In the next step we have to recover the translation

that fits both sets. It seems an easy task, simply applying traditional ICP algorithm over patches as if they were points. But it doesn't work, because we can not forget that the orientation of planes is a very important information. Without orientation we can match two perpendicular planes by proximity. In addition, scene main directions are determinant to obtain the translation. In fact, it is impossible to recover translation if scene planes are aligned in just one direction. With 3DOF we need that planes are distributed, at least, along two directions into the scene.

From these premises, the modified ICP algorithm is similar to the one used in orientation registration but in spite of reaching the angle of rotation between sets, now we compute the translation vector as the weighted sum of vector from one point of the planar patch (usually the center of the patch) to the central point in the other plane in each match. Thus, the weight equation changes as follows:

$$w_i = \gamma \left( e^{-\frac{\alpha_1^2}{\sigma_1^2}} + e^{-\frac{d^2}{\sigma_2^2}} \right) \quad (3)$$

where  $\sigma_1 < \sigma_2$  (empirically  $0.1\sigma_2$  works well) because we have already fit the orientation and we do not want to weight matches of planes with different orientations.

But this algorithm is still not working well in all the situations. For example, let us imagine our robot into the middle of a straight corridor. When this robot performs a movement forward in the direction of the corridor, planar patches extracted from lateral walls of the corridor at each pose fit perfectly without any transformation whereas the wall at the end of corridor does not fit and its transformation represents the movement of the robot. Nevertheless, as corridor wall patches fit perfectly, their weights in the ICP algorithm are maximum and the algorithm can not reach the global minimum.

In order to address it, we propose the following approach: first, we compute the two main directions by Principal Component Analysis over the normal of both sets after orientation registration. After that, in each iteration of ICP we look for the translation in just one of the main directions alternatively. In this case it is very important to remark the necessary changes to compute the translation. Now, each match does not provide the vector between the centre of the patches. Now, Let  $\mathbf{v}$  be the vector from the centre of one plane to its projection over the other plane in a match and let  $\mathbf{d}$  be the direction in which we want to compute the translation, the resulting vector  $\mathbf{u}$  is computed as the projection of  $\mathbf{v}$  over  $\mathbf{d}$ :  $\mathbf{u} = \text{proj}_{\mathbf{d}} \mathbf{v} = \frac{\mathbf{v} \cdot \mathbf{d}}{|\mathbf{d}|^2} \mathbf{d}$  where  $\mathbf{v} \cdot \mathbf{d}$  is the dot product. Thus, let  $\delta = \frac{|\mathbf{u}|}{|\mathbf{v}|}$  be the contribution of the match in the direction  $\mathbf{d}$  the weighted equation is modified as follows:

$$w_i = \gamma \left( \delta \left( e^{-\frac{\alpha_1^2}{\sigma_1^2}} + e^{-\frac{d^2}{\sigma_2^2}} \right) \right) \quad (4)$$



### C. Improved weighting function

In this section we address how to choose the value of variance  $\sigma$ . We know that if  $\sigma$  has a small value the probability of reaching the global minimum increases but at the same time it reduces the range of modeled movements. On the other hand, if we choose a high value for  $\sigma$ , the range of allowed movements is increased, but it is hard to reach a global minimum. To address it we propose a deterministic annealing for managing both orientation and translation registration along ICP iterations. At the beginning of the algorithm  $\sigma$  takes a high value in order to model any kind of movement. As the algorithm iterates, the value of  $\sigma$  is reduced periodically in order to obtain a result as close to the global minimum as possible. With this approach the number of iterations of the algorithm is reduced whereas results are improved.

## V. RESULTS

Data were taken using the 3D scanner described in Section II. We perform our experiments in both indoors and outdoors at the campus of the University of Alicante. Outdoor scenes are formed by a semi-structured scenario formed by alone buildings surrounded by open green areas and trees. Figure 2 (right) shows a campus outdoor point set. It can be noticed that non structured objects such as trees, streetlamps or shrubs produce partial occlusions of the planar surfaces of the environment. 3D images were taken at irregular intervals from 0.5 up to 2.0 meters and 0 to  $\pi/4$  radians. During the experiments, people were walking freely around the robot, which introduce noise into data. The approach described in Section 3 is used to extract planar patches for each point set. After that, the resulting planar patch set is used to register robot movement with the planar patch set from the next 3D scene taken by the robot. No odometry is used in all the process. Algorithms are written using Java language and run on a Dual Core at 3GHz. The time necessary to compute the planes of an outdoor scene is under 2 seconds whereas pose registration of two consecutive images takes about 3 seconds.

Figure 6 shows a zenithal view of the reconstruction obtained from our first outdoor experiment. Floor belonging points have been removed for a better visualisation. Straight lines show walls of buildings, whereas circle-like objects are trees. Red line shows the robot path computed with our approach. In this experiment, the robot performs a path that is almost a cycle, i.e. it is not completely closed, of about 40 meters, and takes 30 3D scenes. The measured mean error for this experiment was 0.04 m for position and 0.035 radians for orientation for each single step into the hole movement.

Figure 7 shows the results of another outdoor experiment. This is also a zenithal view and the floor is also removed for visualisation purposes. In this experiment, robot are surrounded by trees that partially hide buildings. The robot performs a straight movement and at the end of the trajectory turns to the right (without movement). 20 3D scenes were taken. This image shows the results of applying our pose registration algorithm. Although outliers are hardly present in

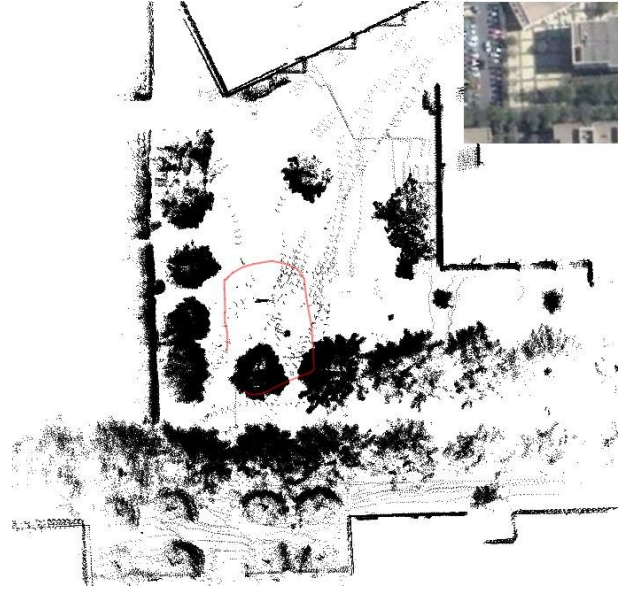


Fig. 6. Zenithal view of the first experiment. Robot movement registration in outdoor semi-structured environment. Red line shows the registered trajectory obtained.

this environment, we perform egomotion with an acceptable pose error of 0.15 meters in position and 0.10 radians in orientation.

In Figure 8 we can observe the results of applying our approach to perform the robot movement registration in an indoor environment. The robot combines translations and turns in each step of the trajectory, which consists of 10 single steps. In this experiment, the environment was hardly dynamic since lots of students were moving through the building. This occurrence can be noticed by the points inside the walls of the corridor. In spite of this dynamic environment we successfully achieve to recover the movement performed by the robot with our egomotion approach. The measured pose error is below 0.04 radians and 0.02 meters.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a method which uses 3D data to obtain the movement performed by the robot. First, we extract 3D planar patches from dense maps obtained from a 3D laser sweeping unit. This process is efficient and allow us to reduce the size of the problem. Then, these patches are used in a modified ICP algorithm, which provides a method to pose registration. This ICP-based method works well in presence of outliers, that is the main lack for original ICP algorithm. Several experiments show the validity of the method for both indoors and outdoors.

Since people were freely moving during experiments, the proposed method has been demonstrated to be valid for dynamic environments. This is because our approach uses planar scene surfaces for achieving pose registration. These surfaces usually remain invariant in time, apart from, for example, doors which can be open or close at different time and thus are considered outliers.



Fig. 7. Second outdoor experiment. Left, zenithal view of the hole reconstruction. Red line shows the registered trajectory obtained. Notice that star-like objects are palm trees. Right, a detailed 3D view of the reconstructed scene.

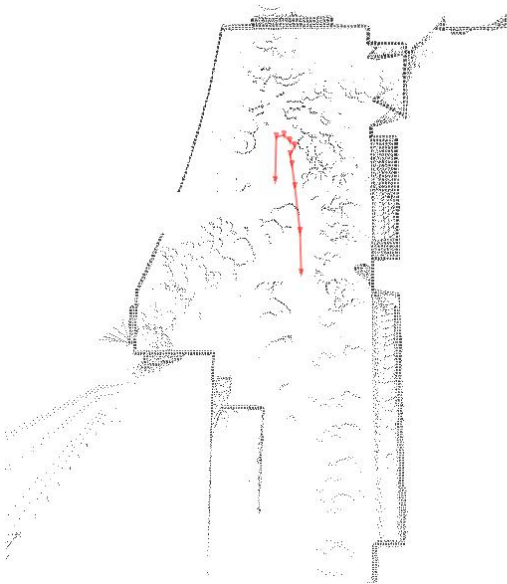


Fig. 8. Zenithal view of an indoor experiment. Red line shows the registered trajectory obtained. Notice that points inside walls are due to students that were moving inside the building during the experiment.

As future work we want to develop a 6 degrees of freedom version of this approach for improving outdoor results. Furthermore, we plan to include the *closing loop* concept in the method, in order to keep on addressing the SLAM problem. Furthermore, we want to explore the inclusion of texture when comparing planes.

#### ACKNOWLEDGMENT

This work has been supported by project GV06/134 from Generalitat Valenciana (Spain) and project DPI2005-01280 from Ministerio de Educación y Ciencia (Spain).

#### REFERENCES

[1] G. Dissanayake, P. Newman, S. Clark, H. Durrant-White, M. Csorba. *A solution to the simultaneous localization and map building (SLAM)*

- problem*. IEEE Transactions on Robotics and Automation. 17 (3). pp. 229-241. 2001
- [2] M. Montemerlo, S. Thrun, D. Koller, B. Webgreit. *FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that probably converges*. In proceedings of the International Joint Conference on Artificial Intelligence, IJCAI, 2003.
- [3] D. Viejo and M. Cazorla *Plane Extraction and Error Modeling of 3D data* 5th International Symposium on Robotics and Automation. Mexico. Agosto, 2006.
- [4] David M. Cole, Alastair R. Harrison, Paul M. Newman. *Using Naturally Salient Regions for SLAM with 3D Laser Data*. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2005
- [5] David M. Cole and Paul M. Newman. *Using Laser Range Data for 3D SLAM in Outdoor Environments*. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2006
- [6] H. Surmann, A. Nuchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. Robotics and Autonomous Systems, 45:181 – 198, December 2003..
- [7] R. Lakaemper and L.J. Latecki. “Extended EM for Planer Approximation of 3D Data”. Proceedings of the 2006 IEEE International Conference on Robotics and Automation. Orlando, Florida. May 2006.
- [8] S. Thrun, C. Martin, Y. Liu, D. Hhnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard. “A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor environments with mobile robots.” IEEE Transactions on Robotics and Automation, 2003.
- [9] R. Triebel, W. Burgard and F. Dellaert, “Using Hierarchical EM to Extract Planes from 3D Range Scans”. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2005
- [10] Weingarten, J. and Siegwart, R. “EKF-based 3D SLAM for Structured Environment Reconstruction”. In Proceedings of IROS, Edmonton, Canada, August 2-6, 2005. (IROS’2005)
- [11] P. J. Besl and N. D. McKay. *A method for registration of 3-d shapes*. IEEE Trans. Pattern Anal. Mach. Intell., 14(2):239-256, 1992
- [12] S. Rusinkiewicz and M. Levoy. *Efficient Variants of the ICP Algorithm*. Third International Conference on 3D Digital Imaging and Modeling (3DIM). 2001.
- [13] M. Martin Nevado, J. Gomez Garcia-Bermejo, E. Zalama Casanova. *Obtaining 3D models of indoor environments with a mobile robot by estimating local surface directions*. Robotics and Autonomous Systems. Elsevier. Vol 48/2-3. pp. 131-143. 2004.
- [14] F. Y. Shih, Shouxian Cheng. *Automatic seeded region growing for color image segmentation*. Image and Vision Computing. Vol 23. pp 877-886. 2005.
- [15] J. Fan, G. Zeng, M. Body and M. S. Hacid. *Seeded region growing: an extensive and comparative study*. Pattern Recogn. Lett. Vol. 26 pp 1139-1156. 2005